#1 ? Команда ls при выводе информации о файлах маркирует флагом "d" файлы:
- прямые ссылки
#2 ? Команда ls при выводе информации о файлах маркирует флагом "c" файлы:
- командные файлы (скрипты)
#3 ? Команда ls при выводе информации о файлах маркирует флагом "b" файлы:
- никакие, этот символ не используется
#4 ? Команда ls при выводе информации о файлах маркирует флагом "m" файлы:
- содержащие метаданные других файлов
#5 ? Команда nice управляет:
- опциями отображения и сглаживания шрифтов
#6 ? Команда gcc – это:
- Giant Crash Comeback
#7 ? В Unix права доступа к файлам и ряду других объектов явно описываются флагами для следующих категорий пользователей:
- владелец, не-владелец
#8 ? Владелец файла всегда, в том числе не имея прав на файл и директорий, может:
- ничего из перечисленного, необходимо вмешательство администратора
#9 ? Возможность удаления файла зависит от наличия:
- прав на модификацию обязательно и файла, и содержащего его директория
#10 ? Флаг SUID в атрибутах исполняемого файла обозначает:
- исполнение потребует смены пользователя (su)
#11 ? Флаг SGID в атрибутах файла обозначает:
- программа, загружаемая из файла, предполагает наличие графического интерфейса
#12 ? Эффективный UID и GID (EUID, EGID) процесса влияют на:
- ни на что не влияют, устаревший элемент в системе
#13 ? Особыми неявно определенными правами доступа к файлам в Unix системах обладает:
- никакие, все права явно описываются битовой маской
#14 ? Пользователь-администратор не обладает:
- правом выполнения двоичных файлов, не принадлежащих root
#15 ? Из перечисленных программ (команд) фильтрами являются:
- top
#16 ? Из перечисленных программ (команд) не являются фильтрами:
- awk
#17 ? Атомарность последовательности операций в контексте параллельного программирования означает:
- критическую опасность для системы в целом
#18 ? "Блокирующий" (blocking) ввод-вывод подразумевает блокировку:
- обязательное объединение операции в один пакет ("блок") с другими схожими операциями
#19 ? В двухуровневой системе выделяются режимы выполнения программ:
- занятость (job mode) и простой (pause mode)
#20 ? В двухуровневой системе выполнение системного вызова предполагает переключение:
- в состояние "зомби"
#21 ? В двухуровневой системе прикладные процессы выполняются в режиме:
- в спящем режиме
#22 ? В двухуровневой системе код функций библиотек, включаемых в адресное пространство прикладных процессов, выполняются в режиме:
- в режиме ядра
#23 ? В иерархически организованной файловой системе файлы идентифицируются:
- хеш-функцией от метаданных файла
#24 ? В состоянии "зомби" (zombie) процесс может:
- сохранять валидным содержимое распределенной ему памяти для других процессов
#25 ? Состояние процесса "зомби" ("zombie") характеризуется:
- изменением EUID на специального пользователя - "сборщика мусора"
#26 ? Завершившийся процесс остается в состояние "зомби" (zombie):
- с момента получения сигнала SIGKILL до окончания его обработки
#27 ? Завершившийся поток (pthread) остается в состоянии "зомби", доступным для считывания его данных:
- до истечения тайм-аута, заданного администратором
#28 ? В число основных функций ядра ОС не включается:
- обработка прерываний, исключений, низкоуровневых событий
#29 ? В Unix для получения разделяемой памяти (shared memory) через отображение файлов (file mapping) необходимо:
- разделяемую память посредством файловых отображений получить невозможно

#30 ? В Unix для открытия существующего файла служит системный вызов:
- FileAttach()
#31 ? В Unix для создания нового файла служит:
- функция PlaceFile()
#32 ? Средствами командной строки новый файл (например, под именем mynewfile) может быть создан:
- cat mynewfile
#33 ? Роль inode в файловой системе:
- хранение карты свободного дискового пространства
#34 ? Inode в файловой системе идентифицируется:
- файловым дескриптором fd
#35 ? Структура inode не содержит информацию:
- о времени доступа к файлу
#36 ? В Unix для удаления существующего файла можно воспользоваться вызовом:
- kill()
#37 ? В результате системного вызова unlink():
- прекращается действие связанного с открытым файлом дескриптора (fd)
#38 ? В файловых системах Unix реальное удаление файлов происходит:
- после подтверждения администратором
#39 ? В Unix используется распределение памяти:
- с использованием фиксированных "банков" памяти
#40 ? В Unix не предусмотрены операций ввода-вывода над файлами:
- разрушающие (операции чтения)
#41 ? В Unix API обращение с запросом к любому драйверу устройства выполняет функция:
- io_control()
#42 ? В Unix API для управления свойствами файла и выполнения операций над ним служит вызов:
- direct_control()
#43 ? В системных вызовах уже открытые файлы (например, в функциях ввода-вывода) идентифицируются:
- указателями на структуру FILE
#44 ? В Unix объект "поток" (thread) идентифицируется:
- только адресом функции потока
#45 ? В Unix объект "поток" (thread) создается:
- из потока ввода-вывода (stream)
#46 ? В Unix объект "процесс" (process) идентифицируется:
- порядковым номером в списке команды top
#47 ? В Unix новый объект "процесс" (process) создается:
- из объектного файла библиотеки
#48 ? Для процессов Unix PID играет роль:
- перманентно неготового устройства
#49 ? В Unix переключение потоков (thread) происходит:
- случайным образом
#50 ? В Unix среди дескрипторов стандартные (стандартно открытые) потоки ввода-вывода имеют числовые значения:
- только 1, 2 и 3
#51 ? В Unix API дубликат дескриптора открытого файла (устройства) может быть получен вызовом:
- copy_fd()
#52 ? Число потоков, одновременно находящихся в активном (running) состоянии:
- не более 4
#53 ? Возможность выполнения конкретным процессом заданных операций над конкретным объектом управляется:
- протоколами
#54 ? Выбор применяемых прав доступа для процесса учитывает:
- дату и время входа в систему
#55 ? Вытесняющий (preemtive) тип многозадачности предполагает:
- вытеснение из системы коммерчески нецелесообразных процессов
#56 ? Дескрипторы созданных/открытых объектов являются в Unix уникальными:
- область видимости определяется программистом
#57 ? Для идентификации объектов pthread_mutex используются:
- указатель на функцию потока
#58 ? Для идентификации уже созданных/открытых объектов System V IPC semaphore служат:
- указатель на объект в адресном пространстве процесса
#59 ? Для идентификации уже созданных/открытых объектов System V IPC shared memory служит:
- дескриптор fd и обязательное имя

#60 ? Для идентификации уже созданных/открытых объектов System V IPC message queue служит:
- дескриптор fd
#61 ? Идентификатор объектов System V IPC формируется:
- выбирается пользователем (программистом) произвольно
#62 ? Попытка повторного (рекурсивного) захвата мьютекса pthread_mutex одним и тем же потоком приводит к:
- поток незапланированно вернется к выполнению
#63 ? Повторный (рекурсивный) захват объекта pthread_mutex возможен в случае:
- невозможен принципиально
#64 ? Для освобождения неоднократно (рекурсивно) захваченного объекта pthread_mutex необходимо:
- освободить объект корректно нельзя, необходимо удалить его
#65 ? Семафор System V IPC представляет собой:
- битовый вектор, поддерживающий поразрядные операции
#66 ? Семафоры System V IPC позволяют выполнять операции:
- инкремент с блокировкой при достижении верхнего предела
#67 ? Семафоры POSIX IPC позволяют выполнять операции:
- атомарный инкремент и декремент связанных семафоров
#68 ? Попытка декремента семафора System V IPC при нулевом его значении приводит к:
- аварийному завершению процесса
#69 ? Попытка декремента семафора POSIX IPC при нулевом его значении может приводить к:
- аварийному завершению процесса
#70 ? Очередь сообщений System V IPC обеспечивает:
- автоматическую конвертацию текстовых сообщений
#71 ? Очередь сообщений POSIX обеспечивает:
- автоматическую переадресацию сообщений
#72 ? Задача взаимного исключения (при взаимодействии вычислительных процессов) состоит в:
- исключении появления процессов-"зомби"
#73 ? Задача непосредственного взаимодействия с внешними устройствами возлагается на:
- прикладные программы
#74 ? Функции драйверов состоят в:
- управлении пользователями
#75 ? Из перечисленных частей образа процесса по умолчанию являются разделяемыми:
- адресное пространство
#76 ? К объектам файловой системы не относятся:
- именованные каналы
#77 ? К числу основных атрибутов процессов Unix не относятся:
- статус (код) завершения процесса
#78 ? Командный интерпретатор (shell) выполняется:
- с приоритетом реального времени
#79 ? Командный интерпретатор (shell) является:
- модулем ядра
#80 ? "Консоль" (текстовый терминал и клавиатура) для операционной системы является логическим устройством такого типа:
- тип задается администратором
#81 ? Критерием выбора планировщиком Linux очередного потока для активизации являются:
- прикладная задача, выполняемая данным потоком
#82 ? Поток, находящийся в критической секции, не должен:
- досрочно покидать секцию
#83 ? "Неблокирующий" (non-blocking) ввод-вывод, в отличие от блокирующего, подразумевает отсутствие блокировки:
- пользователя в системе в случае критической ошибки
#84 ? Обмен данными (достаточно большого объема) между процессами может осуществляться посредством:
- объектов "барьер"
#85 ? Обмен данными между потоками одного процесса не может быть реализован посредством:
- сокетов
#86 ? Особенностью блочных устройств является:
- автоматическая защита от взаимных блокировок
#87 ? Перевод потока из активного (running) состояния в состояние готовности (ready) в системах с вытесняющей многозадачностью происходит по инициативе:
- пользователя – администратора системы
#88 ? Перевод потока из активного (running) состояния в состояние ожидания (wait) в системах с вытесняющей многозадачностью происходит по инициативе:
- других потоков процессов пользователей

#89 ? При изоляции адресных пространств прикладных процессов совместно используемыми бывают:
- ничего из перечисленных
#90 ? При изоляции адресных пространств прикладных процессов совместно используемыми без контроля со стороны самих процессов могут быть:
- сегменты инициализированных данных
#91 ? При асинхронной организации ввода-вывода:
- поток, инициировавший операцию, не может обращаться к объектам синхронизации
#92 ? При мультиплексированном вводе-выводе:
- инициируется множественная параллельная обработка одних и тех же элементов данных
#93 ? Следующий объект может использоваться для синхронизации между потоками разных процессов:
- разделяемая память POSIX shared memory
#94 ? Следующий объект не может использоваться для синхронизации между потоками разных процессов:
- сигнал
#95 ? Состояние процесса "ожидание" ("wait") характеризуется:
- ожиданием удаления из системы завершившегося потока
#96 ? Стандартным (стандартно открытым) потокам ввода-вывода каждого процесса соответствуют:
- глобальные константы – указатели (pointer)
#97 ? Точкой входа в стандартное консольное приложение Unix служит функция:
- app_main()
#98 ? Физическая страница памяти может отображаться в адресное пространство:
- только процесса-владельца и его клонов
#99 ? Флэш-накопитель для файловой системы является логическим устройством такого типа:
- тип задается администратором
#100 ? Для создание именованного канала служит системный вызов:
- attach()
#101 ? Для создания неименованного канала служит системный вызов:
- attach()
#102 ? Для создания именованного канала служит команда:
- такая возможность не предоставляется
#103 ? Средствами командной строки неименованный канал может быть создан:
- pipe program1 program2
#104 ? Управляющие символы <, >, >> в командной строке shell служат для:
- модификации приоритетов в символической форме
#105 ? Барьеры pthread_barrier позволяют:
- приостанавливать одиночный процесс на фиксированное время
#106 ? Код возврата процесса доступен:
- только порожденным процессам посредством канала pipe
#107 ? Процессы-"демоны":
- ограничены приоритетами только холостого хода
#108 ? Результат выполнения системного вызова fork():
- обращение к системе контроля версий проекта
#109 ? При клонировании процесса вызовом fork() новый процесс не наследует от процесса-родителя:
- управляющий терминал
#110 ? Результат выполнения вызовов (функций) exec**():
- инициирование заранее определенного обработчика события
#111 ? Системный вызов syscall() позволяет:
- зарезервировать номер системного вызова для регистрации функции пользователя
#112 ? Функция ожидания сигнала wait() обеспечивает:
- приостановку на заданное время
#113 ? Функция ожидания сигнала waitpid() обеспечивает:
- приостановку до завершения инициализации процесса
#114 ? Функция ожидания сигнала pause() обеспечивает:
- приостановку до получения сигнала SIGCHLD от любого из порожденных процессов
#115 ? Классический сигнал Unix позволяет передать адресату:
- дополнительные "пользовательские" данные
#116 ? Под "ненадежностью" классических сигналов Unix понимаются:
- отсутствие встроенных средств контроля тупиков в обработчиках
#117 ? Обработка полученного сигнала может состоять в:
- автоматической переадресации сигнала
#118 ? Среди видов обработки сигналов не предусмотрено:
- вызов пользовательского обработчика

#119 ? В Unix доставка сигналов адресату происходит:
- только после переключения текущего пользователя
#120 ? Сигнал может быть адресован:
- только процессам-"демонам"
#121 ? Системный вызов signal() служит для:
- имитации обращения к обработчику сигнала с целью отладки
#122 ? Системный вызов kill() служит для:
- удаления учетной записи пользователя
#123 ? Функция raise() служит для:
- восстановления состояния после обработки сигнала
#124 ? Сигнал SIGHUP служит для:
- извещения о значительном превышении процессом лимита ресурсов
#125 ? Сигнал SIGCHLD служит для:
- обмена данными с порожденным процессом
#126 ? Сигнал SIGTERM является:
- извещением о коде завершения процесса
#127 ? Сигнал SIGKILL является:
- извещением об аварийном завершении процесса
#128 ? Сигнал SIGSEGV служит для:
- требования перейти на "вторичный" виртуальный канал
#129 ? Среди сигналов Unix штатно не могут быть перехвачены и игнорированы:
- SIGURG
#130 ? Действующее в конкретный момент времени назначение обработчиков сигналов носит название:
- специализация
#131 ? Для безусловного завершения процесса служит сигнал:
- SIGINT
#132 ? Для завершения процесса с возможностью обработки этой ситуации служит сигнал:
- SIGQUIT
#133 ? Для извещения об ошибке вычислений служит сигнал:
- SIGTERM
#134 ? Для извещения об ошибке обращения к памяти служит сигнал:
- SIGHUP
#135 ? Для извещения о некорректной инструкции ЦП служит сигнал:
- SIGXCPU
#136 ? Для извещений о произвольных "пользовательских" событиях зарезервированы сигналы:
- SIGUTL
#137 ? Сокетам с типом SOCK_DGRAM в рамках стека протоколов TCP/IP соответствует протокол транспортного уровня:
- ARP
#138 ? Сокетам с типом SOCK_STREAM в рамках стека протоколов TCP/IP соответствует протокол транспортного уровня:
- ICMP
#139 ? Сокеты потокового типа позволяют использовать с ними для обмена данными функции:
- перенаправления потоков ввода-вывода
#140 ? Поиск строк, удовлетворяющих шаблону, в потоке или файле обеспечивают фильтры:
- только egrep
#141 ? Из перечисленных утилит (интерпретаторов) и фильтров не поддерживают регулярные выражения:
- grep
#142 ? В командах vim, sed и т.п. первым 10 строкам входного файла (потока) соответствует адрес (кавычки только обрамляют строковые литералы):
- "0..9"
#143 ? Управляющий символ '\U' в команде замены 's' редактора sed обеспечивает:
- представление символа 'U', который иначе является метасимволом
#144 ? Управляющий символ '\L' в команде замены 's' редактора sed обеспечивает:
- выравнивание влево
#145 ? В регулярных выражениях произвольной букве латинского алфавита соответствует конструкция:
- [A..z]
#146 ? В регулярных выражениях одной десятичной цифре соответствует конструкция:
- [0,9]
#147 ? В регулярных выражениях одной шестнадцатеричной цифре соответствует конструкция:
- [0x#]

#148 ? В регулярных выражениях одному знаку арифметической операции соответствует конструкция:
- [+*-/%^]

#149 ? В регулярных выражениях одному необязательному символу "X" соответствует образец (кавычки только обрамляют строковые литералы):
- "[X]"

#150 В регулярных выражениях конструкция "A.B" соответствует подстроке (кавычки только обрамляют строковые литералы):
- содержащей десятичную дробь с точкой-разделителем

#151 В регулярных выражениях трехкратному повтору найденного образца соответствует квантификатор (кавычки только обрамляют строковые литералы):
- "^3"

#152 В регулярных выражениях «якорем» начала строки служит (кавычки только обрамляют строковые литералы):
- символ "<" в начале выражения

#153 В регулярных выражениях «якорем» конца строки служит (кавычки только обрамляют строковые литералы):
- символ ">" в конце выражения

#154 Десятичное число с обязательными двумя знаками после точки можно было бы описать регулярным выражением (кавычки только обрамляют строковые литералы):
- "[-+]?[0-9]*\.[0-9]^2"